NSA-Vol.9 NO.09007

論文

失敗しない引き継ぎを考える

小野　克一郎　（株式会社ミクロスソフトウェア　システムソリューションズ BU）
E-mail:ono@micros.co.jp

要旨：“引き継ぎ”はシステムに携わる者にとって身近な存在である。システムが続く限り業務の引き継ぎも続く。しかしながら引き継ぎに関する学術的な情報は非常に少ない。引き継ぎは軽視されていると言える。

著者が携わったプロジェクトの引き継ぎでも多くの問題が発生した。分析の結果、原因は“職務と情報の属人化”、並びに“引き継ぎ自体の管理方法”であることが分かった。著者のチームで実践している取り組みを例に挙げ、失敗しない引き継ぎについて考える。

キーワード：引き継ぎ、情報管理、属人化、プロジェクト運営

Considering the Successful Handover

Katsu-ichiro Ono (Micros Software, Inc.)

Abstract: “Handover” is close to system developers. The process is executed as long as system continues. The academic resources on the subject are, however, scarce, which is likely to be considered less serious.

The author clarified the problems during the handover process and found those factors in “the personal quality of information and duty” and “the control techniques”. “Successful handover” shall be considered with concrete examples attempted in his project team.

Key Word : handover, information management, personal quality, project management

目次

1. はじめに 2
2. 引き継ぎと機会の増加について 3
2.1. 定義 3
2.2. 概観 3
2.3. 機会の増加について 3
2.4. まとめ 4
3. 引き継ぎで生じる問題点の分析 5
3.1. 潜在的な問題 6
3.2. プロジェクト A に見る失敗例 7
3.3. プロジェクト B に見る失敗例 8
3.4. プロジェクト C に見る失敗例 9
3.5. プロジェクト D に見る失敗例 10
3.6. まとめ 11
4. 失敗しない引き継ぎを考える 12
4.1. 情報管理の取り組み 12
4.1.1. チームの紹介 12
4.1.2. 属人化を防止する 13
4.2. “プロジェクト”としての引き継ぎ 16
4.2.1. プロジェクト E に見る成功例 16
4.3. 課題 19
4.3.1. チーム内での引き継ぎプロセス標準化の試み 19
4.4. 結論 19
5. おわりに 20
参考文献 21

1. はじめに

著者は、予てシステムの開発、保守、運用業務に従事しているが、著者にとって 2008 年は“引き継ぎ”の年であったと言える。立場、規模、状況はそれぞれ異なるが、約 1 年間の間に 10 を超えるプロジェクトの引き継ぎに携わったからである。

業務の引き継ぎを重ねるうちに、引き継ぐ内容やプロセスに起因する様々な問題点が明るみになった。問題にうまく対処できずに、顧客に良くない印象を与えてしまったこともあったに違いない。逆に引き継ぎを成功させていれば、受注できた案件もあったのではないかとも感じる。

引き継ぎはシステムが続く限り避けて通れないプロセスである。また、著者の周囲でも引き継ぎが意図した通りにできなかった結果“プロジェクトが火を噴いた”といった話は良く耳にするものである。にもかかわらず、引き継ぎを体系化した情報はほとんど見当たらない。本来もっと注目されるべきテーマではないだろうか。

本論文では、引き継ぎ時に生じる主要な問題の原因を分析し解決策を考察する。また、その解決策の中から著者自身がチーム内で実践している取り組みを紹介し、その効果を検証する。

2. 引き継ぎと機会の増加について

本章では、初めに引き継ぎの概観を確認する。また、著者の引き継ぎ機会がなぜ増加したのかを考察する。

2.1. 定義

本論文における“引き継ぎ”とは、ソフトウェア産業の業務において、プロジェクト内のある役割から離れる者と、後任としてその役割に就く者との間で交わされる業務内容に関する情報伝達のためのコミュニケーションであると定義する。

また、本論文における“引き継ぎ”の適用範囲は、社内受託での開発、保守、運用業務に限定する。理由は、著者の携わった引き継ぎが、全て自社内での作業であるため、さらには、セキュリティ強固に敏感な客先で行うそれとは考え方、方法、制約などが本質的に異なる場合があるためである。

2.2. 概観

引き継ぎは、システムの開発、保守、運用業務において多様な局面で発生する。いわゆる 2007 年問題[1]として見られた大規模プロジェクトでも、リーダがメンバを兼任しているような小規模プロジェクトでも発生する。また、プロジェクト内部での移動でも、プロジェクト外部への異動でも発生する。さらには、社内受託開発、客先常駐型受託開発、アウトソーシング、人材派遣など、契約形態にも関係なく発生するものである。

そもそも、一般にシステムのライフサイクルに比べて、開発メンバの在任期間のほうが短いため引き継ぎは必然的に発生すると考えることもできる。たとえ開発担当者が現場から離れてしまったとしても、システムが続く限り保守、機能改善、障害対応、マイグレーションやリプレイスは続けていかなければならないのである。

このように、引き継ぎは日々の業務において身近なものだと考えることができる。

2.3. 機会の増加について

なぜ、著者にとって引き継ぎの機会がこれほどまでに増えたのであろうか。理由の 1 つは、著者自身がチームリーダになったことにある。当チームに適した案件がある場合、主にチームリーダが見積もりや調査を行うため、結果として引き継ぎや業務習熟が必要となる。

また、客先常駐者が増加したことも理由の一つである。セキュリティや効率の追求か

[1] 2007 年における団塊の世代の一斉退職に伴い、金融機関等の根幹業務を支えるメインフレームの保守を行える人材が企業に存在しなくなり、経済に悪影響が出るのではないかという懸念から生じたもの。その後、製造業を中心に技術、ノウハウの継承が問題視された。

ら、以前のように大規模開発を社内で請け負うことが困難になっており、多くの人員が社外での作業を行う傾向にある。その結果、自社内で比較的小規模の PJ を担当している著者が、社内のプロジェクトに関する問い合わせや調査の依頼を受ける機会が増加した。

さらには、技術者の入れ替わりが比較的激しい IT 業界において避けられない問題であるが、前任者の異動や退職が重なったことも理由の 1 つである。

上記の 3 つは、人的あるいは組織的な影響によるものであるが、下のようなビジネス動向による理由も考えられる。

1. 当社がかつて開発したシステムで、ソフトウェア、ミドルウェア、ハードウェア機器等のバージョンアップやリプレイスの案件が相次いだ
2. 新規システムの構築より既設システムの改修で要件実現を好む傾向が見られた

1.は、システムが長年に渡る稼働を経た結果、機器の老朽化、保守期限の満了、データ量の増加などに伴う性能改善の必要が高まっているため、新しい技術へ移行しようというものである。つまり、然るべきリハビリのタイミングを迎えたということである。また、それをきっかけとして兼ねてからペンディングとされていた改造要件が合わせて持ち上がることも多く見られた。

2.は、昨今の景気の悪化により、顧客がユーザとの予算採りの折り合いが悪くなっているため、高価な新規システムの構築ではなく安価で実績のあるシステムの改修を提案している結果だと推測される。そして、今後もこうした傾向は続くのではないかと思われる。

2.4. まとめ

引き継ぎはシステムに携わる者にとって身近なもので、誰もが避けて通れないものである。また、2.3.で見たように今後も既存システムの改造案件などが持ち上がることで、引き継ぎに携わる機会があるのではないかと予見している。

引き継ぎが適切に行われなければ、システムの安定稼働は期待できない。次章では引き継ぎに見られる問題点の洗い出しを行う。

3. 引き継ぎで生じる問題点の分析

本章では、引き継ぎの潜在的な問題と、著者が実際に担当したプロジェクトの引き継ぎで見られた問題を取り上げ、分析と考察を行う。

表 3-1 に本論文で参考とするプロジェクトの紹介を行う。

PJ	分類	業務内容	特徴
A	開発	Mobile 端末のアプリケーション開発	・ Mobile 端末を利用した研究開発である ・ プロジェクトが佳境を迎える時期に開発当事者が退職することになり、急遽引き継ぎの必要があった ・ 必要なドキュメントは一通り揃っている
B	保守	データ管理、予約	・ 顧客データ管理と Web による予約機能が組み合わさったシステムである ・ 当社が全機能を開発したが 10 年近くも前の話らしい ・ 改造を重ね保守を継続していたが、新規構築か継続運用かの岐路に立っていた ・ 当時の担当者は残っていない ・ ドキュメントが不十分で詳細設計書などは存在しない
C	開発	交通情報管理	・ 道路、気象情報をブラウザの地図上に表示するシステムである ・ 導入拠点がいくつもあり分散しているシステムや DB が連携しているため規模が大きく、当社で開発したのは一部分である ・ システム稼働から現在までに機能拡張などの改造を何度か行っている ・ 直接の担当者は残っていない ・ 最低限の資料はサーバに保存されている
D	運用	調査結果の集計	・ 集計システムを利用して調査結果を集計する運用業務であり、当社が不慣れな部類の業務である ・ 大規模であるため担当者が多く、人海戦術に頼らざるを得ない性格を持つ ・ プロジェクトの稼働時期が読めないことを始め、諸々の理由で人の入れ替わりが激しく、引き継ぎにも多くの苦労があった

表 3-1 本論文で紹介するプロジェクト

3.1. 潜在的な問題

■コストの問題

[分析]

そもそも、システムに携わる多くの人が、引き継ぎの失敗が原因で苦い経験をしたことがある[2]にも関わらず、“引き継ぎ”自体が軽視される傾向にあるのではないか。著者の部署によく見られる引き継ぎのプロセスも下のように簡素なものであることが多い。

1. 前任者と後任者が打ち合わせを行う
2. 前任者がドキュメントでのレクチャを行う
3. 後任者はドキュメントを元に仕様理解を行い、質問事項をまとめておく
4. 前任者による Q&A を行う
5. その後はメール等で質問を行う

主な理由は時間とコストであろう。後任者の調整もあるため適切な時期に適当な人材のアサインを行うことは容易ではない。結果として、後任者の経験やスキルに関係なく、レクチャ内容、プロセス、期間が同じという理に適わないことも起きているのである。また、このような検討不十分の引き継ぎを行ってしまうと、後から生じる問題により却ってコストが高くなる場合が多いのも事実である。

[考察]

引き継ぎは元来コストが掛かるものだと認識を改めるべきではないだろうか。それには、引き継ぎを単なるイベントとしてではなく、1 つのプロジェクトとして考えるべきである。

一般にプロジェクトの開始段階では、Q(品質)、C(コスト)、D(スケジュール)の管理や予想されるリスクの対策を計画するものである。これを引き継ぎに当てはめるならば、例えば、次のような施策が必要ではないだろうか。

- 品質管理：継承する内容の検討、作成する成果物の確認、目標レベルの設定を行う
- コスト管理：工数を監視する
- スケジュール管理：キックオフからクロージングまでの作業工程を管理する
- リスク管理：後任者が辞めた場合やスキルミスマッチ時のフォロー策を考慮する

引き継ぎにもこのようなプロジェクト同様のマネージメントが肝心であると考える。

[2] 著者の調査では 10 人中 8 人が該当した。

3.2. プロジェクトＡに見る失敗例

駆け出し社員である a は、退職を控えた b 先輩より机上での引き継ぎを受けた。a は、雄弁な b が説明するプログラムの処理内容をイメージできたため、b の退職後も特に危機感はなかった。

ところが実際の開発に入ってみると、ソースコードの質が悪く b にしか理解できないような作りとなっており、到るところに潜在バグが埋め込まれている状態であった。気づいた時には既に遅く、技術力に定評のある社員 c の力も借りて連日の徹夜でなんとか最低限の要求を満たしたが、顧客からクレームを受け、本プロジェクトは失墜する結果となってしまった。

■後任者の技術レベルが現在の担当者よりも低いという問題

[分析]

システムに途中から参加するのだから、後任者の技術レベルや経験は前任者と同等以上であるのが理想である。ところが実際は人的資産の問題もあり、適材適所のアサインができないのが現実である。この場合は、“引き継ぎ”というより寧ろ“教育”レベルの指導が必要となるため、机上だけの引き継ぎでは、後から問題が生じるリスクが高い。

[考察]

このケースでは、前任者と共同で作業を行う OJT 形式での引き継ぎを行うべきであった。OJT で実務経験を積むことにより、業務フローの理解を始め、必要とされる知識や技術の習得が速まり、さらに担当者がドキュメントに書き忘れがちなノウハウの継承なども行えるため有益である。この方法はアジャイル開発を採用する多くのプロジェクトで採用されており効果を上げている。

■退職を控えた社員による引き継ぎの問題

[分析]

一般に退職を控えた社員のモチベーションは低いものである。「お世話になった会社と同僚のために」と最善を尽くしてくれる退職者もいるが、残念ながら極めて稀である。後から判ったことであるが、b はドキュメントの内容漏れや懸念点などを、達弁を振るうことでその場凌ぎを行っていたのである。

[考察]

このケースでは、所属長などマネージャクラスを巻き込み、退職者の有給休暇消化も考えた計画と、責任範囲を明確にした適切な引き継ぎ内容を練る必要がある。また、担当者の退職後は全くフォローを受けることもできなくなる場合も考えられるため、退職予定者と良い人間関係を保つ人材を巻き込むことも必要である。

3.3. プロジェクトＢに見る失敗例

長年に渡り当社が開発と保守を請け負ってきたが、顧客の経営統合の頃合いにより不要機能の切り離し、スピード改善、帳票出力の改善といった要件が持ち上がった。最終的には相見積により、新システムの構築かバージョンアップによる継続運用かの判断がなされることとなり、著者のチームが引き継いだ。

詳細設計書が存在しない上に開発経験者もいない状態[3]であったため、取扱説明書を頼りに動作確認とソースコードの解析による仕様理解を進めた。

調査を進めるうちに、スピードの問題はシステム起動時に全データを構造体に読み込んでいることが原因の一つと考えられた。しかし、何故そのような構造になったのかという経緯が分からない上、改造した場合の影響範囲も見切ることができず、性能改善の具体的な数値などを提示することができなかった。

また、顧客に現在利用している帳票ツールの導入経緯を尋ねたところ、「御社が選んだんじゃないか！」と叱責を受けたこともあったと聞いている。

結果的に、長年培ったシステムは当社から離れることになってしまった。

■情報自体がないという問題

[分析]

必要な情報が残っていないということは、後任者にとって想像を絶することであるが、長年継続しているプロジェクトでは不思議と耳にすることがある。考えられる理由は、バックアップ管理に不備があった、退職者がプロジェクト自体の引き継ぎを忘れた、他会社からプロジェクトを受け継ぐ際に失われた、元々存在しない、などであろう。

[考察]

担当者は時間をかけて作業を行う以外に術はなく、必然的に工数は膨れ上がる。後々のことを考慮してこの機会に情報を整備しておくことが理想であるが、稼働の高い状況ではその場凌ぎの作業に手一杯となってしまいがちである。状況の似ている当社のパッケージ開発の例を見ても、過去に何人もの担当者が存在したが、情報の整理がほとんどなされておらず、担当者が替わる度に同じドキュメントの捜索や同じ不明点の調査を繰り返し行っているような状態である。

こうなってしまうと、もはや解決策はない。この教訓から、現在進行しているプロジェクトが同様の状態にならないように、情報の管理を見直す必要があるのではないか。

[3] このケースはむしろ「引き取り」と呼んだ方が相応しいかもしれない。

3.4. プロジェクトＣに見る失敗例

機器のリプレイスや改造要望がいくつか持ち上がり、著者が見積もりを行うことになった。また、顧客からは(ついでにというわけではないが)、現地で発生しているという障害に関する調査をお願いされた。以前に開発した担当者は残っておらず、過去に少し携わっていたという社員からの間接的な引き継ぎとなった。

サーバに資料は残っているが、過去の対応別にプロジェクトフォルダが枝分れしており、ドキュメントもバラバラになっている。改版履歴などは管理されておらず、どれが最新のドキュメントかさえ判別がつかない状態であった。

過去の資料のみを頼りに、半ば化石発掘に近い形で仕様理解を行ったが、現在の機器構成や導入経緯、仕様書とソースの不一致など不明点が後を絶たず、とても適確な見積もりを行える状況ではなかった。また、そのような状態であるため、障害調査に対しても迅速な対応を取ることができなかった。

結果として、当社が牽引してユーザと直接プロジェクトを進行させてほしいという顧客の要望も、断らざるを得なくなってしまった。

■情報の更新がないという問題

[分析]

これまで改造によるバージョンアップを何度か繰り返していたが、対応時期もバラバラであったため、その節々での引き継ぎがうまくできていなかったのだと思われる[4]。また、そのような状況でスタートせざるを得ないため、作業時間が増加しドキュメントの管理と更新にまで手が回らなかったことも原因であると推測される。

[考察]

システムの稼働から数年たってもこのような案件を紹介されるのは、顧客が経験のあるベンダに先のノウハウを活かした質の高い仕事を期待しているからである。このような形で過去の資産を手放してしまうことは残念であるし、会社全体の信用を下げることにも繋がりかねない。

開発終盤になり機能追加、仕様変更などが増えると仕様書やマニュアルの更新を怠るケースは多い。しかし、いくら情報が存在していても、利用したいときに手軽に利用できなくては情報も価値がない。プロジェクト B の例と同様に情報の管理を見直す必要がある。

[4] 加えて、顧客からシステム全体の更新を反映した資料や稼働開始後の改造履歴などを提供されていなかったことも原因の一つだと思われる。

3.5. プロジェクトＤに見る失敗例

経験者が手薄となり、大大的な引き継ぎを行うことになった。政治や社会情勢に左右される季節労働的性格のプロジェクトのため、必要なタイミングでの人材調達が難しく、引き継ぎに掛かる時間とコストも問題となった。

当社は日頃システムの開発と保守を中心に請け負っているため、運用業務自体に不慣れであった。当初は単純な作りの集計システムや操作マニュアルを見て、誰にでもできる作業であると認識していたが、実際は経験がかなり左右する内容であった。

結果として顧客の期待する応対ができず、実際に新しいメンバで本番運用を経験するまでは、悪い印象を与えることが多かった。

■属人化[5]による問題

[分析]

運用業務の性格上、議事録や業務フローなどのマニュアルはバイブル的存在であるため、他の開発プロジェクトと比べると管理状態は良い。しかし、問題は担当者の頭の中に曖昧な形でしか残っていない類の情報である。業務の多様な局面で、どのような対応が望まれるか、どのような経緯で問題が発生し如何にして解決したのか、と言った情報は、引き継ぎ時に文書化をお願いしたところで、その場で思い出すのは難しいものである。結果的に、退職した経験者に頼らざるを得ず余計なコストをかけることになってしまった。

[考察]

人材の固定化により職務と情報が属人化していたケースである。担当者が不在になる場合などを想定すると、属人化は大きな潜在リスクとなる。後に、引き継ぎ失敗の教訓を活かしダブルキャスト制や担当のチーム化といった戦略を組んだことは、属人化を防ぐことに非常に効果的であった。

実業務を通した OJT による引き継ぎが難しいプロジェクトであるがために、担当者は日頃から作業背景、検討や判断の経緯など些細な個人レベルでの情報も記録しノウハウを共有化する作業プロセスを確立すべきである。

運用業務だけではなくシステム開発や保守業務は属人化しやすい面がある。しかし、このようなチーム化戦略やプロセス改善の試みを取り入れることで大部分は回避できるのではないだろうか。

---

[5] “属人化”とは、人材が固定化し、組織で共有すべきノウハウや情報管理が特定人物に占有され、集団での事業遂行に支障をきたすことである。組織に所属し顧客にサービスを提供する形態の職務においては致命的である。“属人化”の反対は“標準化”と定義されることがある。

3.6. まとめ

本章では、まず潜在的な問題としてコストの問題を取り上げ、引き継ぎへの管理意識を高める必要があるとした。また、著者が経験したプロジェクト引き継ぎの失敗例より、引き継ぎ時に生じる顕著な問題を洗い出し分析と考察を行った。

次章では、これらの問題にどのように対処すれば良いのかを考える。

4. 失敗しない引き継ぎを考える

本章では、3 章で見たプロジェクト引き継ぎの教訓より学び、失敗しない引き継ぎとはどのようなものかを考える。

引き継ぎで失敗しないためには、リスクを減らすことである。著者は、引き継ぎ時に見られる問題への対処法を大きく 2 つに分類した。

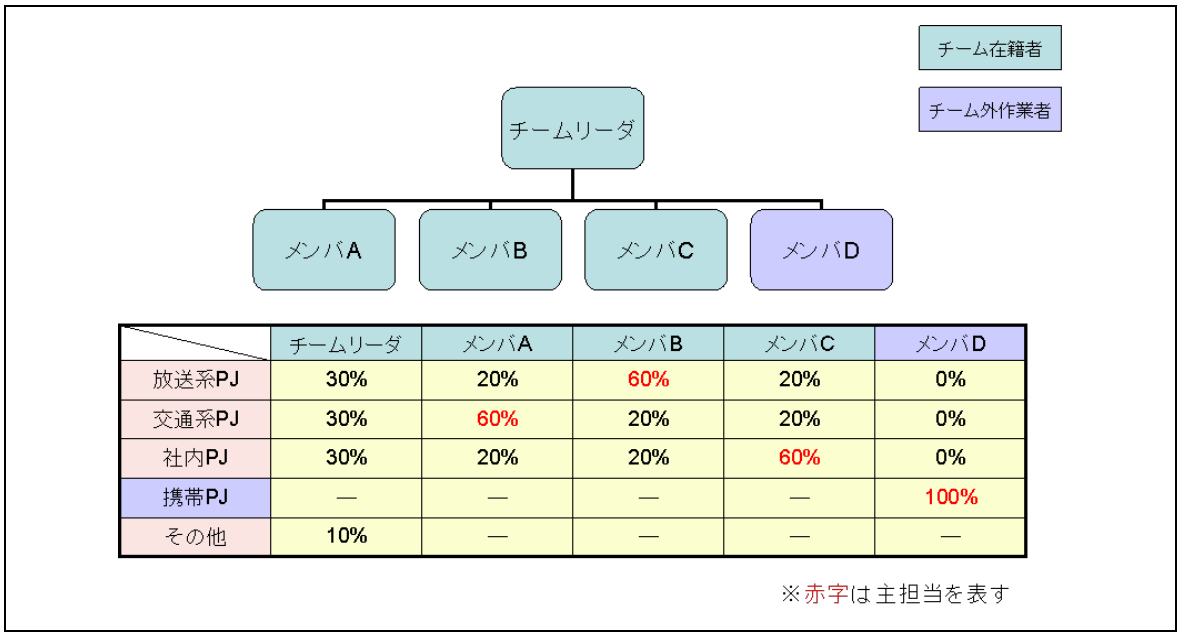
1. 情報を管理(保存、更新、共有)し、現在進行中のプロジェクトの属人化を防ぐこと
2. 引き継ぎを“プロジェクト”として考え QCD とリスクを配慮した運営を行うこと

これらの視点から、著者がチーム運営で行っている具体策を紹介しその効果を検証する。また、本論文のテーマである「失敗しない引き継ぎ」について結論付ける。

4.1. 情報管理の取り組み

4.1.1. チームの紹介

著者のチームは 5 名の編成で、Web 系開発、オブジェクト指向開発を中心に 2~3 のプロジェクトを担当している。図 4-1 に当チームの編成と作業分担を示す。



	チームリーダ	メンバA	メンバB	メンバC	メンバD
放送系PJ	30%	20%	60%	20%	0%
交通系PJ	30%	60%	20%	20%	0%
社内PJ	30%	20%	20%	60%	0%
携帯PJ	—	—	—	—	100%
その他	10%	—	—	—	—

※赤字は主担当を表す

図 4-1 チーム編成と作業分担

チームの特徴として次のようなものが挙げられる。

- プロジェクトは小規模のものが多く、いずれも 1~2 人月程度のボリュームである。
- メンバ D の例に見られるように、会社、組織の状況によっては一時的な他チームへの異動があるため、メンバは一部流動的である。
- メンバは各自が主担当を持っているが、チームリーダは主担当を持たず、管理面、技術面の全般的なサポートとその他の管理作業を行っている。
- 顧客とのやり取りは、主担当またはチームリーダが行っている。
- 社内在籍者は原則としてチーム内の全てのプロジェクトに携わっている。

4.1.2. 属人化を防止する

ここで当チームが積極的に実践している情報に関する試みを紹介する。

■情報保存の試み

メンバは、担当者に依存しがちな情報を記録しプロジェクト管理サーバの所定の場所に格納することを義務付けられている。これらの記録は必要があれば随時更新を行う。

また、打ち合わせ時には必ず簡単な議事録を残す。議事録はサーバの所定の場所に格納され、メンバは課題の経過や個人のタスクをいつでも参照できるようになっている。

これらの文書は書式にはこだわらずメモ書き程度での記録とする。

例えば、プロジェクトの課題管理票や工数管理票では、フォーマットの見直しを図り従来さほど使用されていなかった備考欄のスペースを大きく設けた。備考欄には課題の発生した背景、お客様とのやり取りの内容や反応、解決の見込み、その他気づいた事などを書き込むことになっている。

また技術に関する記録では、調査メモ、設計メモの保存を実施している。調査メモには、日付、調査内容、試みた手順と結果、懸念点、代替案と可能性、設計メモには、日付、設計思想、検討経緯などをそれぞれ含めることになっている。

この類の記録は顧客向けの正式な文書ではないため、時間をかけて質の良いものを作成することには重きを置いていない。これらは、過去の引き継ぎで問題となった「引き継がれなかった」情報であり、ロスを防ぐことが第一の目的[6]であるためである。

次の 2 つの図(4-2、4-3)に、当チームで使用している工数管理票と調査メモの例を紹介する。これらは上記のポイントを意識したドキュメントとなっている。

[6] 実際は担当者自身が作業内容を整理できるという利点もある。

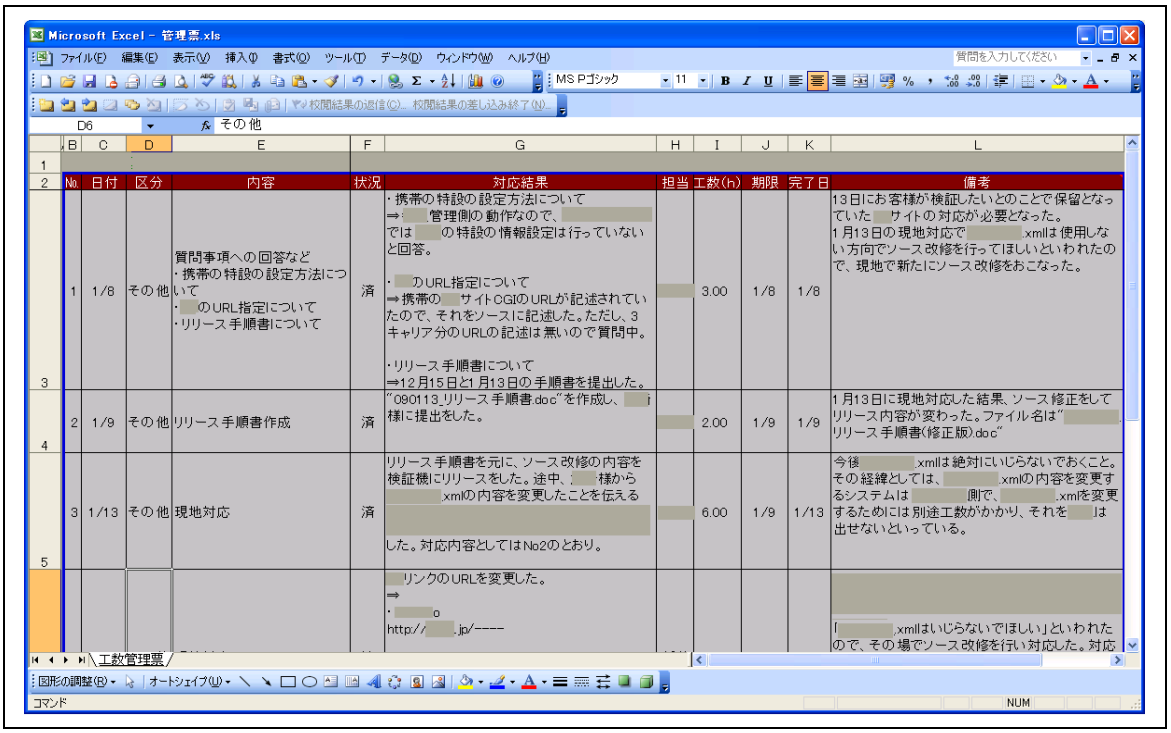


図 4-2 工数管理票の例

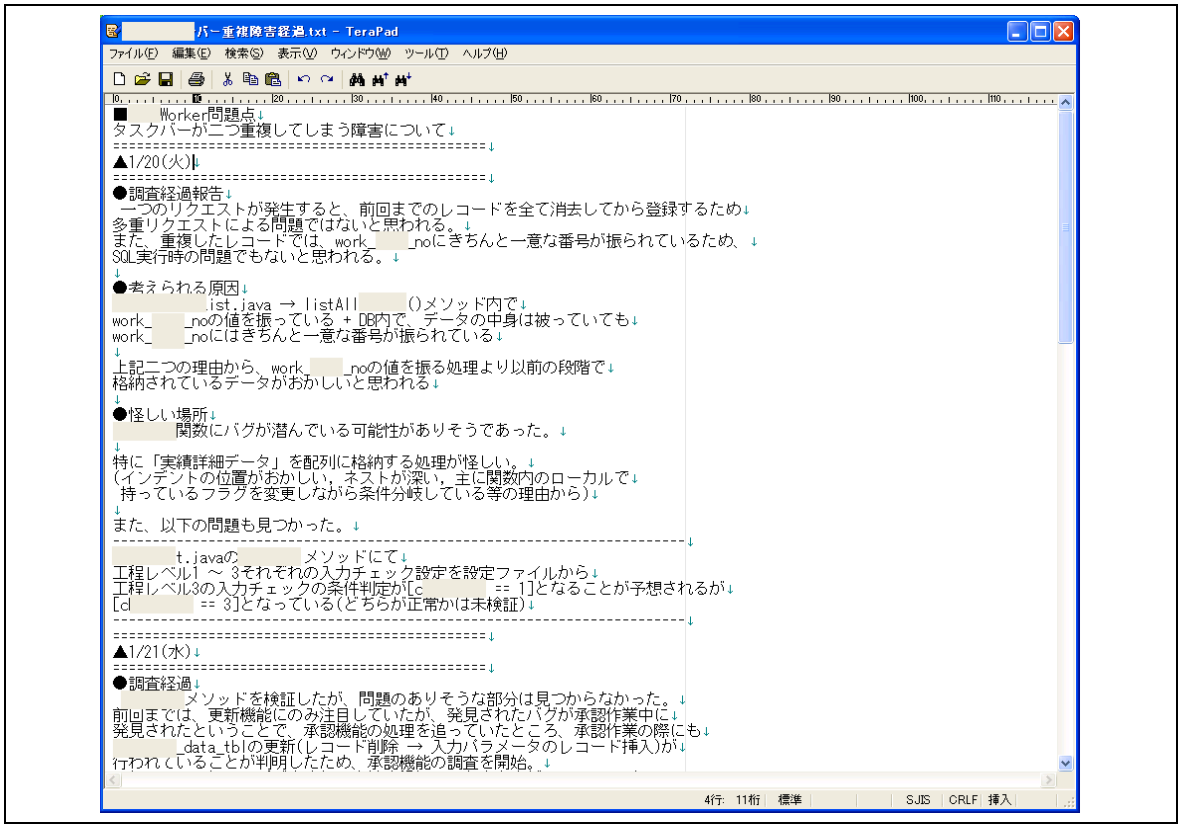


図 4-3 調査メモの例

■情報共有の試み

主担当は設計と実装の中心人物となり、それ以外のメンバは打ち合わせやレビューへの参加、テストの実施などが中心となる作業分担性を適用している。

チームリーダは、プロジェクト管理サーバ内のドキュメント(上で見たものも含む)を定期的にチェックし、毎日の朝会時[7]には状況や進捗の確認を、打ち合わせ時には詳細の報告と議論を行い情報の共有を図っている。

このような体制をとることで、主担当以外も最低限システムの仕様と技術の概要を抑えられる。そのため、テストをメインの実装担当者以外で行えるという品質に影響するメリットもある。メンバもソースのコーディング作業を行うことはあるが、基本的に軽めの内容に限っている。そのためタスクスイッチ[8]の問題も現状見られない。

設計検討などの打ち合わせやレビューはミーティング形式で実施しているため、参加者は誰でも積極的に発言が行えるような雰囲気を心がけている。主担当者に属しているお客さんとのコミュニケーションで得た情報はこの場で全メンバに展開する。

各自が得意とする技術や観点から知恵を絞って議論する結果は強いチーム力となり、単なる技術の継承に留まらず、より品質の良いシステム構築するための情報共有の場となっている。

■考察

以上、情報の管理に関する試みは、急な体制の変更などで担当者がいつ現場を離れることになっても業務に支障が出ないようにする“属人化”へのリスク対策である。簡単に言えば、日頃から引き継ぎを意識し、前任者在任中に後任者を養成する“標準化”への取り組みということである。

チームメンバが流動的であるというのは当チームに限ったことではない。3.1.では引き継ぎの一般的な問題として時間コストとがあることに触れたが、その原因の多くはこのような日々の業務プロセスを怠っているからであると考えている。上記の試みにより実際に引き継ぎが必要となった場合、費やす工数が著しく削減可能であると考えている。

7 当チームの朝会は、問題の詳細に触れないスタイルとなっており 15 分の時間厳守となっている。

8 複数業務担当時の作業切り替えに問題となる脳(短期記憶量)の負荷。コーディング時に特に高いと言われる

4.2. “プロジェクト”としての引き継ぎ

ここで、実際に著者のチームが経験したプロジェクトの引き継ぎを紹介する。表 4-1 に当該プロジェクトの紹介を行う。

PJ	分類	業務内容	特徴
E	開発	交通系制御システム	・ 数年前に当社が新規案件として開発を行った交通系制御システムである ・ 初期メンバである主担当(図 4-1 のメンバ A)が他部署の PJ に参入することとなり、残りのメンバが正式に引き継ぐことになった ・ ドキュメントの保存、更新状態は良い

表 4-1 実際に引き継ぎを行ったプロジェクト

4.2.1. プロジェクト E に見る成功例

本プロジェクトでは、以前より各メンバが仕様理解、設計などのレビューへの参加、テスト、また主担当サポートの元で簡単なソースコードの実装に携わってきた。

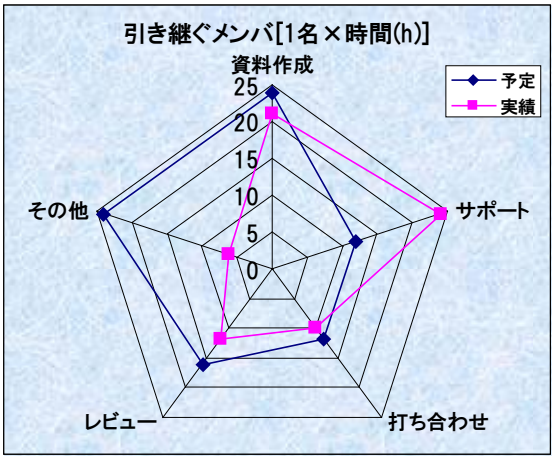
良くある話であるが､諸事情により想定していた引き継ぎ開始時期が 2 週間程度早まり、また予定していた期間も半分近くになってしまった。

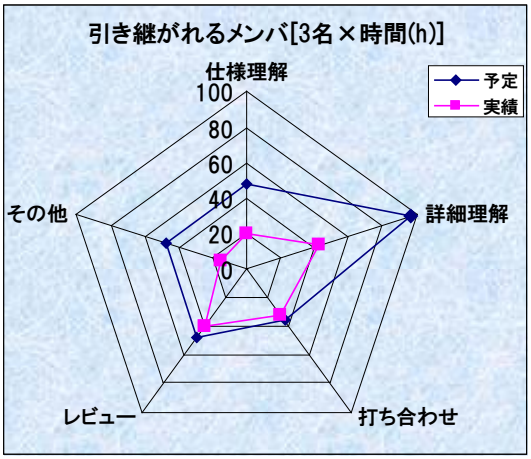
引き継ぎは、他の開発プロジェクト同様にキックオフミーティングを行い、QCD 管理に加えリスク管理を実行した。主担当者と共に作業を行う OJT を一部採用して作業の効率化を狙った。進捗と工数は毎日の朝会を通じて管理し、問題がある場合は早めに前任者のサポートを受けるなどの処置を施した。レビューや打ち合わせは原則全員参加とし、発生した課題に対しては都度担当と期限を割り振った。引き継ぎは、内容と目標レベルの達成を確認するクロージングミーティングを経て完了とした。結果として引き継ぎはスムーズに終了し、残りのメンバは現在も問題なく業務を遂行している。

■考察-1(引き継ぎ期間の短縮について)

充分な期間を設けて内容の濃い引き継ぎを運営することが理想ではあったが､結果として予定工数の短縮という不測の事態にも対応することができた。要因は、無駄を減らしポイントを絞った合理的な引き継ぎが行えたためである。

図 4-4 は、最初に計画していた引き継ぎにかかる工数(予定工数)と期間調整後の工数(実績工数)を、引き継ぐメンバ、引き継がれるメンバ別にレーダーチャートで表したものである。表中の「仕様理解」とはドキュメントの理解、「詳細理解」とはソース解析など技術面の理解、また、「その他」とは、検討不測や漏れなどが原因で発生すると思われる予期せぬ作業を行うためのバッファ、つまりリスク分の工数である。


引き継ぐメンバ[1名×時間(h)]
資料作成
25
20
15
10
5
0
予定
実績
その他
サポート
レビュー
打ち合わせ


引き継がれるメンバ[3名×時間(h)]
仕様理解
100
80
60
40
20
0
予定
実績
その他
詳細理解
レビュー
打ち合わせ

■考察-2(管理について)

著者は、今回の引き継ぎをチーム全員が担当する“プロジェクト”として捉えた。そのため、プロジェクト同様にキックオフミーティングを実施した。主な内容は、目的、役割分担、継承範囲の確認、引き継ぎ開始と完了フェーズでの理解度の目標設定、成果物の確認、リスク対策であった。これらを考慮した上で入念な工程の立案を行った結果、メンバは他のプロジェクト開始時と同様にモチベーションの高い状態で作業を完遂できたと考える。

■考察-3(前任者の成果物について)

属人化している情報の洗い出しを行った結果、前任者が作成すべき成果物は次のようなものとなった。

成果物	内容
作成資料一覧	各作成資料の説明 「いつ」、「だれが」、「何のために」、「何を」を含む情報
作業フロー説明書	見積もり依頼が来てからリリースを行うまでの業務の流れ図
環境説明書	環境構築、ツール説明、顧客支給品等の情報
テスト手順書	シナリオテストの進め方 テストツールについての説明
リリース手順書	「バージョンを変更したか？」など項目毎のチェックシート
今後の動向予想	今後改修が入ると予測される内容と、主な修正クラスの説明
引き継ぎ完了報告書	当事者情報、引き継ぎ内容と課題、目標達成度、今後の連絡先など
議事録群	Q&A による返答内容： サーバフォルダ構成、ソース管理方法、設計思想、開発時の工夫、お客様情報、今後の見通し、過去に起きた問題点など
過去のメール	サーバの所定位置に TEXT 形式で配置

表 4-2 前任者の成果物一覧

これらのドキュメントは、引き継ぎ期間中に行われた何度かのレビューを通して内容の見直しが図られ、情報の漏れが少ない良質のものとなった。また、上に挙げた成果物

一覧は、内容は多少異なっていても、他の開発プロジェクトで引き継ぎが発生する際に成果物の目安として使用できるのではないかと考えている。

■考察-4(実業務の経験)

今回の引き継ぎが終盤に差し掛かったころ、顧客から障害の改修依頼が舞い込んだ。普段であれば気が進まないところではあるが、主担当に代わって解析、修正、テスト、障害票の記入、リリースまでの一連の流れを経験できるため良い機会であった。ここでも OJT を採用したが、主担当はサポートに専念し、引き継がれるメンバのみで対応を行った。その結果、今後の作業に対する不安が一掃された。また、主担当者自身も実作業を行うことで、机上の引き継ぎでは忘れがちな生きた情報を思い出し、文書に反映することができた。

以上、日頃の情報管理への取り組みと、引き継ぎを“プロジェクト”として入念に運営した結果が、失敗のない引き継ぎを実行できた要因であったのではないかと考える。

4.3. 課題

4.3.1. チーム内での引き継ぎプロセス標準化の試み

紹介したように、情報管理と運営管理を怠らなければ、当チーム内のプロジェクトの引き継ぎに関する限り大きな問題はないと考える。しかし、実際に開発当初からのメンバが離れてしまうほどの大掛かりな引き継ぎを行ったのは今回が初めてである。また、引き継ぎ後は以前ほど大きな作業が発生していない状況であるため、まだ明るみになっていない問題が存在する可能性もある。今後いくつかの引き継ぎを経験した段階で、前任者が作成する成果物とその管理方法、マネージャが運営として行うべき作業などは標準化を行い、フローやチェックリストを整備したいと考えている。

4.4. 結論

本章では、引き継ぎ時に見られる問題への対処法として、情報管理によるプロジェクトの属人化防止、引き継ぎを“プロジェクト”として考えた運営、の 2 点に着目し、著者のチームでの取り組みと実際に経験した引き継ぎを成功例として紹介した。

成功例として紹介したプロジェクトも、4.3.の課題でも触れたように、水面下ではまだ課題を残していると思われる。プロジェクトが続けば日々増えていくドキュメントの更新管理という難しい課題も浮上するであろう。

しかしながら、100%漏れのない引き継ぎなど存在しない。そう考えれば、普段から引き継ぎを意識した作業を進め、前任者の在任中に後任者を養成しておくこと、また、“プロジェクト”と変わらぬ意識で引き継ぎに臨むことは、引き継ぎで失敗しないための要件だと断言できるのではないか。

5. おわりに

本論文のテーマであるシステムの“引き継ぎ”は、この一年の業務で著者が一番困ったことは何かを思い返した結果、取り上げたものである。

いざ、書こうとしたものの、情報の少なさに驚かされた。学術的な書籍の出版は確認できず、Web 上でも纏まった形で取得できる情報は限られていた。稀に保守業務に関する引き継ぎは、IT 関連のムックなどでトピックとなることがあるようであるが、それでも体系的な情報は得られなかった。いくら引き継ぎが多様であるからとはいえ、もう少し注視を浴びるべきである。

著者のチームでは「チーム内プロジェクトの情報共有」を今期の目標の一つとして掲げているが、目標を設定した時点では情報共有の取り組みが、引き継ぎに効果があるとは想像さえしなかった。論文のテーマと身近なチーム運営が重なり、良い発見となった。

今後は、課題にも掲げた引き継ぎの標準化への取り組みに挑戦したいと考えている。

参考文献

(1) Jim Highsmith、Agile Project Management、日経 BP 社、2005 年 6 月

(2) 小野克一郎、森淳一郎「研究案件におけるチーム化の効果と展望」、当社イントラ 2008 年 2 月

(3) 広兼修、プロジェクトマネージメント標準　PMBOK 入門、オーム社、2005 年 11 月

(4) 2007 年問題に関する記事群
http://www.itmedia.co.jp/enterprise/special/0601/2007/